initial

Reads input files broadcasts, and allocates/initializes global variables.

[called by: focus, globals.] [calls: .]

Input namelist

The *focusin* namelist is the only input namelist needed for FOCUS running. It should be written from the file *example.input*. Here are the details for the variables.

- `IsQuiet = -1`
  Information displayed to the user
  -1: more details & update unconstrained cost functions;
  -1: more details;
  0: essential;
  1: concise.
- `IsSymmetric = 0`
  Enforce stellarator symmetry or not
  0: no stellarator symmetry enforced;
  1: periodicty enforced;
  2: fully stellarator symmetry enforced.

- `case_surface = 0`
  Specify the input plasma boundary format
  0: general VMEC-like format (Rbc, Rbs, Zbc, Zbs), seen in rdsurf;
  1: read axis for knots, seen in rdknot; (not ready)
- `knotsurf = 0.2`
  Minor plasma radius for knototrans, only valid for `case_surface = 1`
- `ellipticity = 0.0`
  Ellipticity of plasma for knototrans, only valid for `case_surface = 1`
- `Nteta = 64`
  Poloidal resolution for discritizing the plasma
- `Nzeta = 64`
  Toroidal resolution for discritizing the plasma

- `case_init = 0`
  Specify the initializing method for coils, seen in rdcoils
  -1: read the standard *coils.example* file;
  0: read FOCUS format data in *example.focus*;
  1: toroidally spaced `Ncoils` circular coils with radius of `init_radius`;
- `case_coils = 1`
  Specify representation used for the initial coils, seen in rdcoils
  0: using piecewise linear representation; (not ready)
  1: using Fourier series representation;
- `Ncoils = 0`
  Number of coils initilized, only valid for `case_init = 1`
- `init_current = 1.0E6`
  Initial coil current (A), only valid for `case_init = 1`
- `init_radius = 1.0`
  Initial coil radius (m), only valid for `case_init = 1`
- `IsVaryCurrent = 1`
  Keep coil currents fixed or not, overriden by example.focus
  0: coil currents are fixed;
  1: coil currents are free;
- `IsVaryGeometry = 1`
  Keep coil geometries fixed or not, overriden by example.focus
  0: coil geometries are fixed;
  1: coil geometries are free;
- `NFcoil = 4`

Number of Fourier coefficients for coils, only valid for `case_coils = 1`*, overriden by* example.focus

- `Nseg = 128`
  Number of segments for discritizing coils, only valid for `case_coils = 1`*, overriden by* example.focus

---

- `IsNormalize = 1`
  Normalizing coil parameters or not
  0: keep raw data (normalized to 1.0);
  1: currents being normalized to averaged absolute current, coil geometry parameters being normalized to major radius;
- `IsNormWeight = 1`
  each constraints normalized to initial value or not
  0: keep raw value for constraints;
  1: $w = w/f_0$ weights normalized to the initial values of each constraints;
- `case_bnormal = 0`
  Bn error normalized to $|B|$ *or not*
  0: keep raw Bn error;
  1: Bn residue normalized to local $|B|$;
- `case_length = 0`
  options for constructing coil length constraint, seen in length
  1: quadratic format, converging the target_length;
  2: exponential format, as short as possible;
- `weight_bnorm = 1.0`
  weight for Bn error, if zero, turned off; seen in bnormal
- `weight_bharm = 0.0`
  weight for Bn Fourier harmonics error, if zero, turned off; seen in bmnharm
- `weight_tflux = 0.0`
  weight for toroidal flux error, if zero, turned off; seen in torflux
- `target_tflux = 0.0`
  target value for the toroidal flux, if zero, automatically set to initial Ψ_{ave}*; seen in solvers*
- `weight_ttlen = 0.0`
  weight for coils length error, if zero, turned off; seen in length
- `target_length = 0.0`
  target value (or for normalization) of the coils length, if zero, automatically set to initial actual length; seen in rdcoils
- `weight_specw = 0.0`
  weight for spectral condensation error, if zero, turned off; seen in specwid; (not ready)
- `weight_ccsep = 0.0`
  weight for coil-coil separation constraint, if zero, turned off; seen in coilsep; (not ready)

- `case_optimize = 1`
  specify optimizing options.
  -2: check the 2nd derivatives; seen infdcheck; (not ready)
  -1: check the 1st derivarives; seen infdcheck;
  0: no optimizations performed;
  1: optimizing with algorithms using the gradient (DF and/or CG); seen in solvers;
  2: optimizing with algorithms using the Hessian (HT and/or NT); seen in solvers;
- `DF_maxiter = 0`
  maximum iterations allowed for using Differential Flow (DF); if zero, turned of; seen in descent
- `DF_xtol = 1.000D-08`
  reletive error for ODE solver; seen in descent
- `DF_tausta = 0.000D+00`
  starting value of τ*; usually* 0.0 *is a good idea; seen in descent*
- `DF_tauend = 0.000D+00`
  ending value of τ*; the larger value of* $\tau_{end} - \tau_{sta}$*, the more optimized; seen in descent*
- `CG_maxiter = 0`
  maximum iterations allowed for using Conjugate Gradient (CG); if zero, turned of; seen in congrad
- `CG_xtol = 1.000D-08`
  the stopping criteria of finding minimum; if $|d\chi^2/d\mathbf{X}| < CG\_xtol$*, exit the optimization; seen in congrad*;
- `CG_wolfe_c1 = 1.000D-04`

c1 value in the strong wolfe condition for line search; usually 1.0×10^{-4}*; seen in congrad*;

- `CG_wolfe_c2 = 0.1`

  c2 value in the strong wolfe condition for line search; if one CG step takes too long, try to increase c2, but remember $0 < c1 < c2 < 1$*; seen in congrad*;

  \- -

- `case_postproc = 1`

  specify post-processing options.

  0: no extra post-processing;

  1: evaluate the current coils; more details; (not ready)

  2: write mgrid file; (not ready)

- `save_freq = 1`

  frequence for writing output files; should be positive; seen in solvers;

- `save_coils = 0`

  flag for indicating whether write example.coils*; seen in saving*;

- `save_harmonics = 0`

  flag for indicating whether write example.harmonics*; seen in saving*;

- `save_filaments = 0`

  flag for indicating whether write .example.filaments.xxxxxx*; seen in saving*;